Deep Just-In-Time Inconsistency Detection Between Comments and Source Code

Sheena Panthaplackel (spantha@cs.utexas.edu), Junyi Jessy Li, Milos Gligoric, Raymond J. Mooney

The University of Texas at Austin

Problem
When developers make code changes, they often fail to update comments accordingly. This results in inconsistent comments that lead to time-wasting confusion and vulnerability to bugs.

Data: Mining comment-method pairs from consecutive commits (C, M) with code changes (i.e., M_new ≠ M), from open-source Java projects on GitHub.

Inconsistent: Pretrained and jointly trained systems perform similarly.

Goal: Determine whether a comment is inconsistent, just-in-time, i.e. right before code changes are merged into a code base.

Evaluation

Intrinsic Evaluation
- Liu et al. (2018): 75.8 \( \text{F1} \) 76.3 \( \text{Acc} \)
- Post Hoc SEQ: 63.0 \( \text{F1} \) 60.3 \( \text{Acc} \)
- Just-In-Time SEQ: 81.5 \( \text{F1} \) 82.0 \( \text{Acc} \)
- Just-In-Time GRAPH: 81.4 \( \text{F1} \) 82.0 \( \text{Acc} \)
- Just-In-Time HYBRID: 83.1 \( \text{F1} \) 83.8 \( \text{Acc} \)
- Just-In-Time HYBRID + features: 87.1 \( \text{F1} \) 87.8 \( \text{Acc} \)

Extrinsic Evaluation: Evaluating a comprehensive comment maintenance system which automatically updates a comment (Panthaplackel et al., ACL 2020) if inconsistency is detected by our newly proposed approach.

Training

Update w/ Implicit Detection

Inference

\[ P_{\text{inconsistency label}} = \text{out(Update)} \neq C \]

\[ P_{\text{comment}} = \text{out(Update)} \]

\[ \text{if out(Detection) is Inconsistent: } P_{\text{comment}} = \text{out(Update)} \]

\[ \text{else: } P_{\text{comment}} = C \]

Results in noisy data → Cleaned test sample for more reliable evaluation